FOUNDATIONAL BUSINESS ANALYTICS
LOOP THE LOOP

"LOOPS" - bits of code that iterate, over and over until some condition is met. Along with data types, variables and conditionals, they form the core of any 'imperative' programming language. Master these and most of the hard work in programming is done!

PYTHONIC LOOPS:

1. Create a new program file. Go to IDLE's output window, navigate to the File menu and choose 'New File'. This will give us an empty sheet as normal in which to write a new program script.

2. Type the following code into the window:

```
print("My first FBA loop!")
x = 0
while (x < 5):
   print("Line " + str(x))
   x = x + 1
```

3. Success? Good - note that the variable x was keeping track of how many times we looped, and we incremented it each time the loop ran through. If we hadn't had that line the loop would have run forever (as x would always have been less than zero). Try deleting that line and see your first infinite loop!!

4. Oops. It's going to run forever. Press **ctrl-c** to make it stop!

5. Okay, so this way of using a variable (**x** in this case) to keep track of where you are in a loop is now very old fashioned. And not 'pythonic' at all (seriously, that's a word. Google it). Instead try the following...

6. Update your code to the following:

```
print("My second, more pythonic FBA loop!")
for x in range(5):
    print("Line " + str(x))
```

   Remember that range(5) is just producing the list [0,1,2,3,4] which we are now iterating through, putting each element into x every time we go through the loop, and dropping out when we run out of list elements.

7. Prove this be replacing range(5) with your own list of numbers... (remember a list is just an ordered set of items in [square brackets])

8. Great your code is now more beautiful and pythonic' (If you've coded before forget those old fashioned C and Java style loops - time to modernize!).

THIS TIME ON YOUR OWN - WHILE:

1. Write a While loop that asks the user to input some text, and prints it out onto the screen until they type "**quit**". Remember to use the **input()** command.
2. Adjust your program, so that the words they type in get added to a string with spaces between the words (use an extra variable to store this running log string!)
3. Adjust the program again so the words get appended to a list not an ever growing string.
4. Complete your program by adding a conditional statement that checks if the user typed in the word display. If so, then don't store the word, and instead print out the current list to screen, before continuing the loop.

THIS TIME ON YOUR OWN - FOR:

1. Write a program that prints the numbers from 1 to 10, using a **for** loop and the **range()** function.
2. Adjust your code so the numbers are all printed on one line. This is done as follows:

   ```
   for item in range(1, 11):
       print(str(item), end=" ")
   print()
   ```

   The print statement has a thing called a named parameter that you can use, called end. Normally after a print a newline is added. In this case we've replaced it with a space! Test it out, by trying some other options. Note the final print() - it's not specifically needed here, but once we've come out the loop this ensures we'll start on a newline next time we print something else!
3. Modify your code so that the numbers are printed in the reverse order (from 20 to 1). N.b experiment with the range function to achieve this.
4. Modify the range function so that it prints all the even numbers between 0 and 20 (see: https://docs.python.org/3/library/functions.html#func-range for examples of using the range function in this way)
5. Return the range function to the way it was and modify your code so that it prints all the even numbers between 0 and 100, but this time using the modulus operator (%) to check if the number is divisible by 2 before it's printed to screen
6. BACK TO TRIANGLES! Write a program that will use a for loop to print out the following triangle of numbers:

   ```
   1
   1 2
   1 2 3
   1 2 3 4
   1 2 3 4 5
   ```

```
1 2 3 4 5 6
```

(Tip - use a nested loop to do this!)

7. Your doing great trainee looper. Okay, so this is the big one. Generate the following triangle without using the * symbol anywhere in your code....

```
     1
    1 2
   1 2 3
  1 2 3 4
 1 2 3 4 5
1 2 3 4 5 6
```

8. If you have mastered the above, you are now well on your way. I had to ask for help when I did the above exercise in my **MSc** a *cough* couple of years ago, so don't expect to do it first time.

   Congrats - you have achieved yellow belt business analytics Well, you would if we had belts. We don't. But feel free to buy yourself a yellow belt however. Then you can tell people that's why. I don't recommend this at parties.

YELLOW BELT LOOPING:

Okay, we're going to see how you are getting on when you're not just following instructions. The following exercises will be slightly harder as they will involve a loop and some variable manipulation

1. **Celsius converter**: Write a loop that converts Celsius to Fahrenheit. The program should print a conversion table between Celsius and Fahrenheit. This should provide conversions for from -50 Celsius to 50 Celsius, in increments of 5 degrees, using a for loop.

   Tip: to convert Fahrenheit, multiply Celsius by 9, then divide by 5, then add 32.

   Test your program by checking the results are the same as this page:
   http://www.mathsisfun.com/temperature-conversion.html

2. **Fibonacci**: Write a program that produces the first 10 items in the fibonacci sequence - if you don't know the sequence see the (slightly misnamed obviously):
   https://www.mathsisfun.com/numbers/fibonacci-sequence.html

   The output of your program should look something like the following:
   ```
   1    1    2    3    5    8    13 ...
   ```

3. **FIZBUZ**: This is a classic of learning to code exercises. It's more challenging, and should take you the rest of the time in the lab, and then some. It combines a drinking game with loops and conditionals. Seriously. *What could be more fun. (Don't answer that obviously).*

Ok, your task is to write a program that prints out the FIZBUZ series, 5 numbers on each line, formatted as best you can. The Fizbuz series simply writes the number in the series apart from the following rules:

a. *If the number is divisible by 3 OR contains a 3 replace it with the text "Fiz"*

b. *If the number is divisible by 5 OR contains a 5 replace it with the text "Buz"*

c. *If both the above are true then replace the number with "Fizbuz"*

Output should look something like:

```
1, 2, Fiz, 4, Buz,
Fiz, 7, 8, Fiz, Buz,
11, Fiz, Fiz, 14, Fizbuz,
16, 17, Fiz, 19, Buzz ...
```

Tip: For this I would use the AND, OR, % and // operators for different checks!

Congratulations! If you have not coded before getting to this stage is already a big achievement, as you have mastered the fundamentals of script coding, which is now at the heart of almost all analytics. Pat yourself on the back.

If you have coded before, always strive when you use python to do things in the most efficient way possible, strive for elegance in your code, and write things 'pythonically'. You will see this will his will move you away more and more from the styles of other languages. So keep learning!